DOI: 10.55648/1998-6920-2026-20-2-3-15 УДК 004.056.5

Privacy-preserving intrusion detection in corporate networks via federated and differentially private deep learning

Abdul Qayyum[1], Hamid Idris Mussa[1], Khalil Ibrahim[1], S. V. Bezzateev[2]

[1]Saint Petersburg National Research University of Information Technologies, Mechanics and Optics
[2]Saint-Petersburg State University of Aerospace Instrumentation

Abstract: Centralized intrusion detection in distributed corporate infrastructures (branches, remote offices, and enterprise/industrial IoT) creates privacy and compliance constraints because it requires centralizing sensitive telemetry. We study a privacy-preserving IDS design based on federated learning (FL) with differentially private local training (DP-SGD). Clients train locally on flow-derived features and structured event telemetry, and only model updates are shared with a coordinator for aggregation. We report the privacy budget (ε, δ) using an RDP accountant and evaluate detection quality using Accuracy and Macro-F1. Experiments on CICIDS 2018 and the CERT Insider Threat v6.2 dataset show the expected privacy-utility trade-off: DP training reduces utility compared to non-private centralized learning, and FL with DP typically incurs an additional decrease under heterogeneous (non-IID) client partitions, while remaining practical at explicit privacy budgets (main comparison: $\varepsilon \approx 1.8$ -2.0). We also report SOC-relevant operational indicators: training time, peak memory usage, inference latency, and model size-and compare against Random Forest and XGBoost baselines.

The work was performed within the framework of the state assignment (project FSER-2025-0003).

Keywords: Intrusion Detection, Federated Learning, Differential Privacy, DP-SGD, Secure Aggregation, Insider Threats, Information Security, Corporate Networks, Security Operations Center (SOC).

For citation: Qayyum Abdul, Mussa Hamid Idris, Ibrahim Khalil, Bezzateev S. V. Privacy-preserving intrusion detection in corporate networks via federated and differentially private deep learning [Paper Preparation Manual for Vestnik SibGUTI]. Vestnik SibGUTI, 2026, vol. 20, no. 2, pp. 3–15. https://doi.org/10.55648/1998-6920-2026-20-2-3-15.

The article was submitted: 27.02.2026;
revised version: 16.04.2026;
accepted for publication 16.04.2026.

1. Introduction

The rapid digitalization of corporate operations and the shift toward distributed infrastructures (branch networks, remote offices, and industrial/enterprise IoT) have expanded both the attack surface and the dependence on continuous telemetry [2, 5]. At the same time, privacy control and internal compliance requirements increase the cost and risk of collecting, transporting, and storing network and user-level traces, which may include sensitive personal or commercially valuable information [2, 4]. From an information security perspective, this creates a practical tension between detection performance and minimizing exposure of sensitive telemetry. In practice,

modern security analytics must balance high detection quality with strict constraints on data exposure [2].

Machine-learning-based intrusion detection systems (IDS) are widely used because they can capture high-dimensional traffic patterns and evolving attacks that go beyond static signatures and manually written rules [1]. However, many deployments still rely on centralized aggregation of flow records and host/user activity logs for model training and tuning [2]. This centralization increases the confidentiality risk surface and often conflicts with data-minimization expectations [2, 11]. Even without payloads, metadata and behavioral traces can be sensitive, especially for insider-style monitoring, where the signals may involve access patterns, time/role context, and organizational context [2].

Federated learning (FL) offers a natural alternative: it enables collaborative model training without moving raw telemetry to a single location [3, 4]. In typical FL workflows, clients train locally and share model updates with an aggregation server [3]. However, decentralization alone does not guarantee privacy [6]. Model updates can still leak information about local records and can be targeted by reconstruction or inference attacks under realistic assumptions [7, 10]. Therefore, distributed IDS designs require additional protection that provides quantifiable privacy guarantees against leakage through training artifacts [11].

Differential privacy (DP) addresses this need by limiting the influence of any single training record on the learned model, thereby reducing what an adversary can infer from released updates or outputs [11]. In deep learning, DP is commonly implemented through DP-SGD, which clips per-example gradients and injects calibrated noise, while tracking the overall privacy loss via accounting methods [12, 14]. In addition, the honest-but-curious coordinator threat can be further reduced using secure aggregation, where the server observes only aggregated updates rather than individual client contributions [6, 16].

In this paper, we introduce a privacy-preserving architecture for intrusion and confidentiality-violation detection in corporate networks using federated optimization with differentially private local training. The framework targets heterogeneous client distributions (e.g., departments, branches, and device groups) and supports integration into Security Operations Center (SOC) workflows without centralizing sensitive telemetry [2, 4, 5]. Our contributions are:

1) an end-to-end design with an explicit threat model for FL with DP-SGD and optional secure aggregation [3, 12–14, 16];

2) a feature pipeline that combines flow-derived network telemetry with structured event-based signals and supports non-IID client settings [4, 5];

3) an empirical evaluation of the privacy-utility trade-off under explicit privacy budgets (ε, δ), together with operational metrics relevant for deployment [11–14].

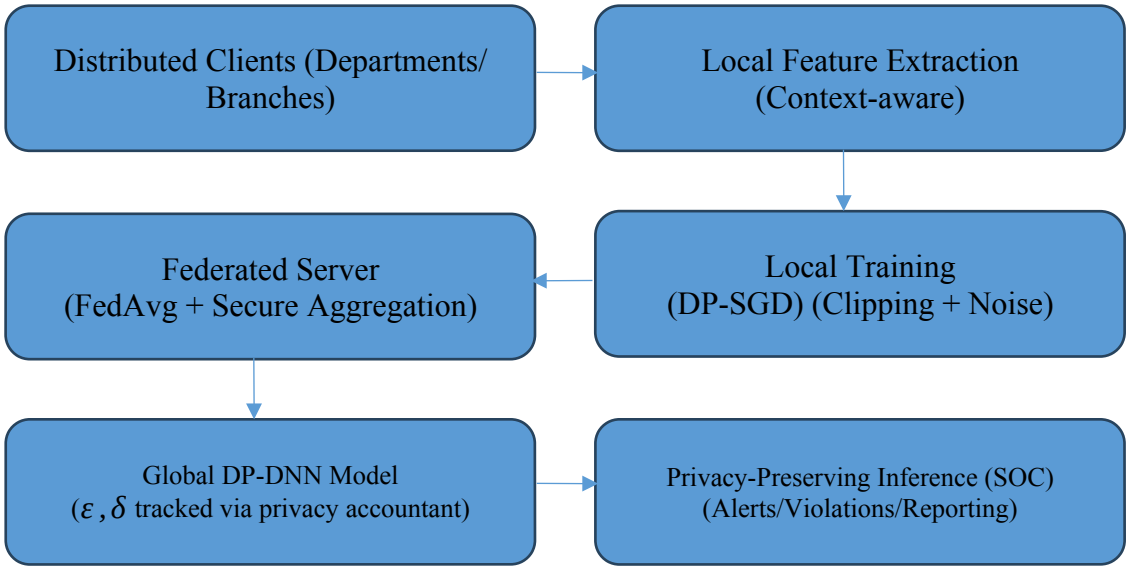

Fig. 1. High-level architecture of the proposed privacy-preserving intrusion detection framework (federated learning with DP-SGD and optional secure aggregation)

The high-level architecture is shown in Fig. 1. We report Accuracy, Macro-F1, training cost, memory footprint, inference latency, and model size together with the privacy budget (ε, δ) [1, 11–14].

2. Related work

2.1. Machine-learning IDS and benchmark datasets

Machine-learning intrusion detection systems (IDS) are widely used because modern enterprise telemetry is high-dimensional and attack patterns evolve faster than static rule sets. In this setting, feature design, dataset representativeness, and evaluation protocol strongly influence reported performance, especially under class imbalance and distribution shift [1]. For network intrusion detection, flow-based benchmarks (CIC-style datasets) are common because they match enterprise monitoring pipelines built on aggregated statistics rather than payload inspection [18, 19]. For insider-threat style detection, network-only signals are often insufficient; structured user-host activity traces are typically used to model access patterns and behavioral deviations, and the CERT insider-threat dataset is a standard reference in this direction [20].

2.2. Intrusion detection with federated learning

Federated learning (FL) has been explored for IDS when policy, regulation, or operational constraints make centralizing raw telemetry unattractive. FL keeps data local and exchanges model updates, which makes collaboration possible without sharing raw traffic logs or host traces [3, 4]. In practice, aggregation-efficient training (e.g., FedAvg) is widely used, but FL-IDS deployments face two recurring issues: client heterogeneity (non-IID data) that degrades the global model, and resource/availability limits at endpoints (compute, bandwidth, intermittent connectivity) [3, 4]. Recent surveys summarize these challenges and discuss architectural and aggregation choices for FL-based detection in security monitoring [4–6].

2.3. Privacy leakage in FL and differential privacy

Decentralization alone does not guarantee privacy: model updates (gradients, deltas) can leak information about local records. Reconstruction and gradient-inversion style attacks show that sensitive samples or attributes may be inferred from updates under certain training regimes [7, 8]. Membership inference attacks further show that an adversary can sometimes detect whether a specific record was used in training, even when exact reconstruction is not possible [9, 10]. Differential privacy (DP) addresses this risk by limiting the influence of any single record on the learned model [11]. In deep learning, DP is commonly enforced via DP-SGD (per-example clipping + calibrated noise), and privacy loss is tracked by composition-aware accounting such as Rényi DP [12–14]. In FL settings, DP can be applied locally at clients to protect record-level information while still enabling aggregation [15].

2.4. Integrity threats: poisoning and backdoors

Beyond confidentiality, FL also introduces integrity threats: a participant may poison training or insert a backdoor so that the global model behaves normally on validation data but fails on targeted inputs [17]. Since the goal of this paper is confidentiality-preserving detection with explicit (ε, δ) reporting, we treat poisoning/backdoors as a deployment risk rather than a fully solved problem. We therefore discuss robustness options (e.g., screening and robust aggregation) as constraints and future work directions, while keeping the paper's main focus on the privacy-utility trade-off (ε, δ vs Macro-F1) and SOC feasibility metrics (cost, memory, latency).

3. Problem statement and threat model

3.1. Detection objective

We consider an enterprise network divided into organizational units (clients) indexed by k. Each client k holds a local dataset D_k derived from (i) flow-generated network traces and (ii) structured event telemetry. Each record has features x and a label y (multi-class intrusion label or a binary anomaly/confidentiality-violation label depending on the task). The goal is to train a global detector f_θ that minimizes expected loss across clients:

$$\min_{\theta} \sum_{k} E_{x,y \sim D_k}\left[L\left(f_\theta(x), y\right)\right].$$

Here $L(.)$ is cross-entropy in the multi-class setting and binary cross-entropy in the binary setting.

Operationally, we target two detection settings reflected by our benchmarks:

1. Network intrusion detection (e.g., scanning, DoS/DDoS, infiltration, botnet-like activity) using flow-oriented features compatible with enterprise monitoring and CIC-style datasets [18, 19].
2. Confidentiality-violation/insider-type behavior using structured event telemetry (user-host events, access patterns, time/role context) consistent with insider-threat benchmarks such as CERT [20].

3.2. Threat Model

We assume an enterprise FL deployment with the following threats and trust assumptions:

1. Honest-but-curious coordinator (server). The coordinator follows the protocol but may attempt to learn sensitive information from client updates. Optional secure aggregation reduces visibility into individual updates by exposing only aggregates [16].
2. Sensitive client data. Raw telemetry remains local, but information may still leak through gradients/updates if no privacy protection is applied [7–10].
3. Adversary querying the deployed model. Under realistic assumptions, an attacker may use model outputs to support inference-style privacy attacks (including membership inference) [9, 10].
4. Optional malicious clients (integrity risk). Some clients may deviate from the protocol (poisoning/backdoors). In this work, robustness to such attacks is treated as a deployment risk and future direction rather than a fully solved component [17].
5. Record-level privacy via DP-SGD. DP-SGD is used to limit what can be inferred about any single training record from released model parameters and updates, with privacy loss tracked by an accountant [12–14].

3.3. Security goal

Our security goal is to reduce leakage of client data through training artifacts (updates) and model outputs, while keeping detection performance practical for SOC workflows. We therefore evaluate (i) detection quality (Accuracy, Macro-F1) and (ii) operational feasibility indicators (training cost, memory footprint, inference latency, model size), together with the reported privacy budget.

4. Methodology

4.1. Overall pipeline

We consider a distributed corporate environment where each organizational unit (e.g., department, branch, site) acts as a client in a federated learning (FL) workflow [3, 4]. Each client performs: (i) local preprocessing and feature extraction from its telemetry, (ii) local training of a detection model, and (iii) transmission of a model update (e.g., parameter delta) to a coordinating server [3, 4]. The server aggregates updates into a global model and redistributes it to clients in subsequent communication rounds [3]. This design avoids centralizing raw network traces and user-

activity logs. When enabled, secure aggregation further limits server visibility by ensuring it can observe only aggregated updates rather than individual client contributions [16]. Differential privacy (DP) is applied during local training to bound leakage through updates and to provide an explicit privacy budget [6, 11–14].

4.2. Normalization and feature engineering

Our representation follows enterprise monitoring practice and CIC-style IDS benchmarks, where detection relies on flow-level statistics rather than payload inspection [18, 19]. We use transport-level features and time-based summaries (e.g., packet/byte rates, inter-arrival summaries), directional indicators, and protocol/flag aggregates [18]. Where available, we incorporate contextual cues (e.g., time-of-day patterns, periodicity, burstiness, coarse role/location labels) to support insider-style telemetry such as CERT v6.2 without using direct identifiers [20].

To handle heterogeneity and client drift in non-IID settings, features are standardized per client using robust statistics when appropriate and aligned to a shared schema across clients [4]. Feature selection is performed in two stages: (i) a filter stage (e.g., mutual information ranking) to remove redundancy, and (ii) a lightweight wrapper stage on a validation split to avoid degrading predictive performance under DP noise [4, 12].

4.3. Training objective and model

Let $f_\theta(.)$ denote the detector that maps a feature vector x to a class label (multi-class intrusion) or a binary label (anomaly/violation). We use a fully connected deep neural network (MLP) for tabular flow/event features, trained with cross-entropy (multi-class) or binary cross-entropy (binary). To keep comparisons fair, the same model family is used across centralized DNN, DP-DNN, and FL+DP-DNN so that observed differences reflect the training setup (centralized vs FL, DP vs non-DP) rather than architecture changes [3, 4, 12].

4.4. Federated optimization (FedAvg)

At communication round t, the server sends parameter θ_t to a subset of client S_t. Each selected client k trains locally on its dataset D_k and returns an update. The server aggregates client updates using sample-size weighting:

$$\theta_{t+1} \sum_k \frac{n_k}{n} \theta_t^k, \quad n = \sum_{k \in S_t} n_k$$

where n_k is the number of training samples used by client k in round t. In scalability experiments (Section 6.4), we vary the total number of clients while keeping the client-selection rule fixed [3, 4].

4.5. Differentially private local training (DP-SGD)

To provide formal privacy guarantees against leakage through model updates, clients use DP-SGD during local training [12]. With mini-batch B, per-example gradients g_i are clipped to an l_2 norm bound C, and Gaussian noise is added before normalization:

$$\overline{g} = \frac{1}{|B|} \left(\sum_{i \in B} clip(g_i, C) + N(0, \sigma^2 C^2 I) \right).$$

Privacy loss is tracked using composition-aware accounting based on Rényi differential privacy (RDP) and reported as (ε, δ) [13, 14]. In all reported results, (ε, δ) denotes the cumulative privacy budget over the full local training procedure rather than a per-round privacy budget. In the main comparison, we report results at explicit privacy budgets $\varepsilon \approx 1.8 - 2.0$ (Table 1) [12, 14], obtained with a fixed noise multiplier $\sigma = 1.2$. For the privacy-utility curve (Fig. 2), we vary the noise multiplier σ while keeping the remaining training procedure fixed, which produces different ε values under the accountant [12, 14].

4.6. Secure aggregation (optional)

Secure aggregation is complementary to DP. It protects at the client-update level by reducing server visibility into individual updates (honest-but-curious coordinator), while DP limits record-level leakage even from aggregated outputs. In this paper, secure aggregation is treated as an optional deployment enhancement; the base experimental results focus on DP-enabled training [6, 16].

4.7. Deployment considerations

The framework supports common enterprise deployment patterns: multi-site FL coordinated by an SOC, hybrid edge deployments where gateways federate local learners, and federated consortia where raw telemetry sharing is infeasible [4]. Because the approach does not require packet payloads and reports explicit privacy budgets, it supports privacy/compliance reporting in addition to operational detection goals [11–16].

5. Experimental Setup

5.1. Datasets and client splits

We evaluate the proposed privacy-preserving IDS design on two benchmark datasets that reflect common enterprise monitoring settings [18–20].

CICIDS 2018 (network intrusion): This dataset represents flow-based intrusion detection using aggregated network statistics (rather than payload inspection) [18, 19]. We treat CICIDS as a multi-class intrusion benchmark and use a flow-feature representation consistent with enterprise monitoring pipelines [18].

CERT Insider Threat v6.2 (confidentiality/insider behavior). This dataset captures structured user-host activity and is used to model insider-style behavioral deviations [20]. In our experiments, we use a 4-class setup: Normal, Data theft, Sabotage, and Misuse, matching the labels reported in Section 6 [20].

Client construction (IID vs non-IID). For federated experiments, we simulate organizational units (clients) using two partitioning regimes [3, 4]:

1. **IID split:** samples are randomly distributed across clients to approximate homogeneous telemetry [3].
2. **Non-IID split** clients are formed to induce heterogeneous label distributions and realistic drift (e.g., time-window or prevalence-based splits), producing skewed local objectives across organizational units [4].

This allows us to study performance under both cooperative and heterogeneous enterprise conditions, and to connect directly to the scalability analysis in Section 6.4 [4].

5.2. Train/validation/test protocol

Each dataset is split into train/validation/test = $70/15/15$. The test split is held out and is never used for tuning. Hyperparameters and model selection decisions are based on the validation split.

For federated learning, client partitioning is applied to the training split only [3]. Validation is used for tuning, and all reported detection metrics are computed on the held-out test split to ensure a consistent and unbiased comparison across centralized and federated settings [3, 4].

5.3. Models and baselines

We compare classical feature-based IDS baselines against neural models under centralized training and privacy-preserving training [1]:

1. Random Forest (RF) and XGBoost as strong non-neural baselines for tabular flow/event features [1].
2. Centralized DNN: trained on the combined training data (centralized learning) [1].

3. DP-DNN (centralized): the same DNN trained with DP-SGD to quantify the privacy-utility trade-off without federation [12–14].

4. FL+DP-DNN: federated training using FedAvg where each client applies

5. DP-SGD locally and the server aggregates updates [3, 12–14]. This is the primary privacy-preserving federated configuration evaluated in Section 6 [3, 12–14].

All neural configurations use the same model family and feature representation so that performance differences can be attributed to the training setup (centralized vs federated, DP vs non-DP) rather than to differences in features or architecture [3, 12].

5.4. Measures of evaluation and system cost

Detection quality is evaluated using:

– Accuracy

– Macro-F1, which is emphasized because intrusion datasets are typically imbalanced and macro-averaging better reflects per-class performance [1].

To assess operational feasibility for SOC-style deployments, we also report:

1. Training time (end-to-end runtime for training),
2. Peak memory usage (maximum RAM consumption during training),
3. Inference latency (per-sample or batch inference time, as reported in Section 6),
4. Model size (storage footprint of the trained model).

Where randomness is involved (e.g., initialization or client sampling), we report results in the form mean ± standard deviation when feasible [3, 4].

6. Results and Discussion

6.1. Trade-off between privacy and utility

Table 1 summarizes detection performance on CICIDS 2018 and CERT v6.2 under centralized training, differentially private training, and federated differentially private training, together with classical baselines (Random Forest, XGBoost) [1, 18–20]. As expected, the centralized DNN achieves the strongest utility because training uses pooled data [3, 4]. When DP-SGD is enabled, performance decreases due to per-example gradient clipping and injected noise during optimization [12]. Under FL+DP-DNN, performance typically decreases further because FedAvg must combine heterogeneous (non-IID) client updates while each client trains with DP noise locally [3, 4, 12, 14].

Even with this reduction, Macro-F1 remains informative for SOC-style triage [1, 2], and the privacy guarantee is communicated explicitly through the reported privacy budget ε [11–14]. This enables a quantifiable privacy-utility evaluation rather than a qualitative privacy claim.

Table 1. Detection performance on CICIDS 2018 and CERT v6.2 under centralized, differentially private, and federated differentially private training

Model	Dataset	Accuracy	Macro-F1	Privacy Budget (ϵ)
Random Forest	CICIDS 2018	0.972	0.963	-
XGBoost	CICIDS 2018	0.975	0.965	-
DNN	CICIDS 2018	0.981	0.974	-
DP-DNN	CICIDS 2018	0.963	0.951	1.980
FL+DP-DNN	CICIDS 2018	0.951	0.938	1.850
DNN	CERT v6.2	0.953	0.940	-
DP-DNN	CERT v6.2	0.938	0.921	1.950
FL+DP-DNN	CERT v6.2	0.927	0.911	1.820

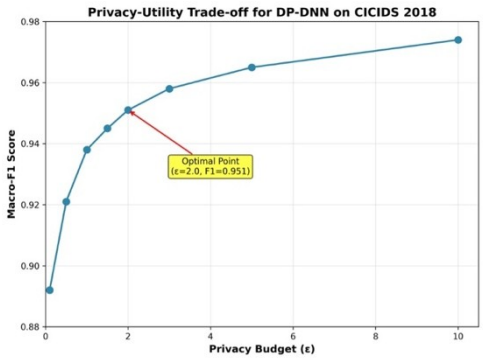

Fig. 2. Privacy-utility trade-off: Macro-F1 versus privacy budget (ε) for DP-DNN on CICIDS 2018

Figure 2 illustrates the expected monotonic privacy-utility trend for DP-DNN on CICIDS 2018: stronger privacy (smaller ε) leads to lower Macro-F1, consistent with DP-SGD behavior [12, 13]. For the curve in Fig. 2, we vary the noise multiplier σ while keeping the remaining training protocol fixed [12–14].

6.2. Dynamics of training under differential privacy

To complement the final metrics, we examine optimization behavior under DP [12]. Figure 3 compares training loss convergence and validation accuracy for standard DNN versus DP-DNN [12]. DP-SGD slows convergence because clipping limits gradient magnitude and injected noise increases gradient variance, which delays early improvements compared to non-private training [12, 13]. After sufficient epochs, both regimes stabilize, indicating that the chosen DP setting remains learnable and produces a usable detector. Figure 3 corresponds to the same DP configuration used in the main comparison in Table 1 [12–14].

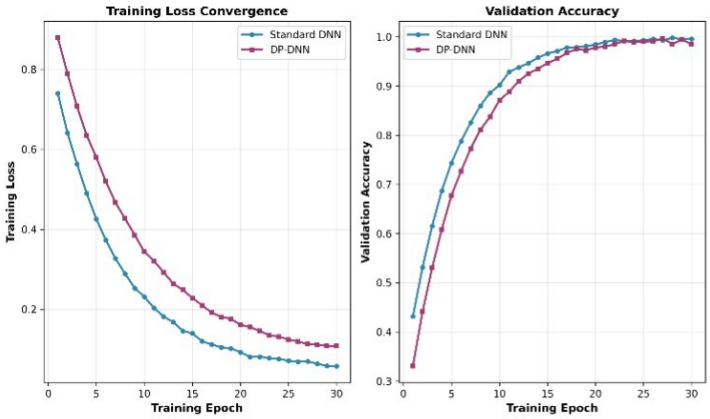

Fig. 3. Training dynamics for standard and differentially private training: a) training loss convergence; b) validation accuracy versus training epoch

6.3. Telemetry-based insider type of class-based behavior

To provide a class-level view of confidentiality-violation and insider-style events, Figure 4 presents the confusion matrix of DP-DNN on the CERT insider-threat dataset (Normal, Data theft, Sabotage, Misuse) [20]. The model separates the dominant Normal class well and captures policy-relevant categories. Most errors occur between behaviorally similar categories, which is expected because activity patterns overlap and DP regularization tends to smooth decision boundaries [11, 12].

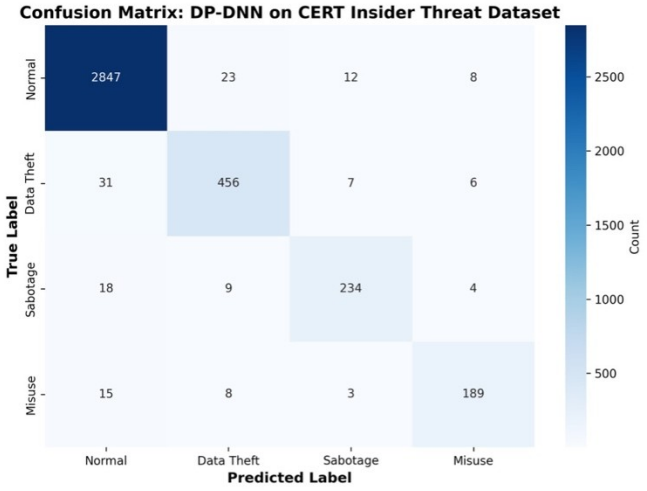

Fig. 4. Confusion matrix of the DP-DNN on the CERT insider-threat dataset (Normal, Data theft, Sabotage, Misuse)

6.4. Heterogeneity versus client expansion

Figure5 evaluates scalability by increasing the number of FL clients under IID and non-IID partitioning [3, 4]. With IID splits, performance decreases only slightly as the client base grows. Under non-IID splits, the decrease is stronger because client drift increases the variance of updates and creates objective mismatch under FedAvg-style aggregation [3, 4]. We evaluate $K = 10, 20, 50,$ and 100 clients, while keeping the client sampling fraction per round fixed across K. The observed trend suggests diminishing returns past a mid-range number of clients unless non-IID mitigation is applied (e.g., personalization, clustered FL, or robust aggregation).

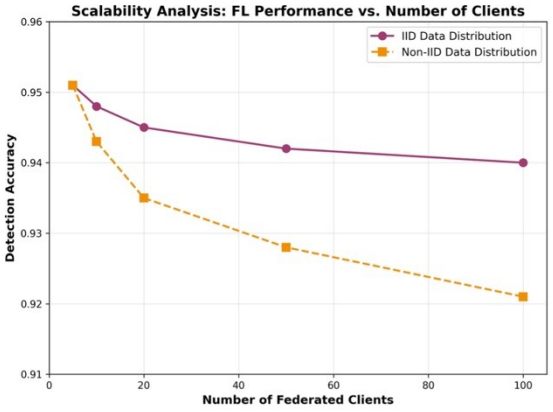

Fig. 5. Scalability under client growth: detection accuracy versus the number of federated clients for IID and non-IID client data partitions

6.5. Productivity and deployment practicability

Table 2 reports SOC-relevant operational indicators: training time, peak memory usage, inference latency, and model size. Tree-based baselines provide low inference latency, although they may be less flexible under evolving traffic patterns compared to neural detectors. DP training increases runtime and memory overhead due to per-example gradient processing and privacy accounting. FL+DP adds additional training time due to repeated local updates and communication rounds. Memory usage is reported as peak RAM observed during training, including dataset in-memory representation and optimizer states. Overall, the overhead is practical for environments where centralizing raw telemetry is restricted, and it makes the privacy-utility trade-off explicit for compliance-oriented deployment [2, 11].

Table 2. Efficiency and deployment feasibility: training time, peak memory usage, inference latency, and model size for baseline and privacy-preserving IDS configurations

Model	Training Time (min)	Memory Usage (GB)	Inference Latency (ms)	Model Size (MB)
Random Forest	12.3	2.1	4.2	45.2
XGBoost	18.7	3.4	5.1	52.8
DNN	45.2	8.9	8.3	124.5
DP-DNN	67.8	12.3	9.1	124.5
FL+DP-DNN	89.4	15.7	11.2	124.5

7. Conclusion

This paper studied a practical privacy-preserving intrusion detection design for distributed corporate environments where centralizing sensitive telemetry is costly or infeasible. We combined federated learning with differentially private local training (DP-SGD) so that clients train on flow-derived and structured event telemetry locally and share only model updates for aggregation, with optional secure aggregation as an additional protection layer against an honest-but-curious coordinator.

Across CICIDS 2018 and CERT Insider Threat v6.2, the results show a clear and expected privacy-utility trade-off. Centralized non-private training achieves the best detection utility, while DP-SGD reduces performance due to clipping and noise injection, and federated training with DP typically incurs an additional decrease under heterogeneous (non-IID) client partitions. At the same time, the privacy guarantees are stated explicitly through the reported privacy budget, making the system suitable for compliance-driven settings where privacy risk must be quantified rather than only claimed.

Beyond detection accuracy and Macro-F1, we reported SOC-relevant operational indicators (training time, peak memory usage, inference latency, and model size). These measurements show that the proposed configurations remain deployable in practice, with predictable overhead introduced by DP training and repeated FL communication rounds. Overall, the results support DP-enabled federated detection as a feasible option for enterprise and industrial networks when telemetry centralization is not acceptable. At the same time, robustness against poisoning and backdoor attacks was not evaluated experimentally and remains a limitation of the present work. Future work will therefore focus on adversarial robustness, stronger mitigation of highly non-IID client distributions, and validation in more realistic enterprise deployment scenarios.

References

1. Buczak A. L., Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 2016, vol. 18, no. 2, pp. 1153-1176. DOI:10.1109/COMST.2015.2494502.
2. Scarfone K., Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). NIST Special Publication 800-94. National Institute of Standards and Technology, 2007. DOI:10.6028/NIST.SP.800-94.
3. McMahan H. B., Moore E., Ramage D., Hampson S., Agüera y Arcas B. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of AISTATS*, 2017. arXiv:1602.05629.
4. Kairouz P., McMahan H. B., Avent B., et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 2021, vol. 14, no. 1-2, pp. 1-210. DOI:10.1561/2200000083.
5. Lazzarini R., Tianfield H., Charissis V. Federated learning for IoT intrusion detection. *AI*, 2023, vol. 4, no. 3, pp. 509-530. DOI:10.3390/ai4030028.

6. Chen J., Yan H., Liu Z., Zhang M., Xiong H., Yu S. When federated learning meets privacy-preserving computation. *ACM Computing Surveys*, 2024, vol. 56, no. 12, pp. 1-36. DOI:10.1145/3679013.
7. Zhu L., Liu Z., Han S. Deep leakage from gradients. In: *Advances in Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019, pp. 14747-14756.
8. Geiping J., Bauermeister H., Dröge H., Moeller M. Inverting gradients - how easy is it to break privacy in federated learning? arXiv:2007.05657, 2020.
9. Shokri R., Stronati M., Song C., Shmatikov V. Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 3-18. DOI:10.1109/SP.2017.41.
10. Nasr M., Shokri R., Houmansadr A. Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. *2019 IEEE Symposium on Security and Privacy (SP)*, 2019. DOI:10.1109/SP.2019.00065.
11. Dwork C., Roth A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2014, vol. 9, no. 3-4, pp. 211-407. DOI:10.1561/0400000042.
12. Abadi M., Chu A., Goodfellow I. et al. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308-318. DOI:10.1145/2976749.2978318.
13. Mironov I. Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 263-275. DOI:10.1109/CSF.2017.11.
14. Wang Y.-X., Balle B., Kasiviswanathan S. P. Subsampled Rényi differential privacy and analytical moments accountant. In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS 2019)*. PMLR, vol. 89.
15. Geyer R. C., Klein T., Nabi M. Differentially private federated learning: a client level perspective. arXiv:1712.07557, 2017.
16. Bonawitz K., Ivanov V., Kreuter B., et al. Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175-1191. DOI:10.1145/3133956.3133982.
17. Bagdasaryan E., Veit A., Hua Y., Estrin D., Shmatikov V. How to backdoor federated learning. In: *Proceedings of AISTATS, 2020*. arXiv:1807.00459.
18. Sharafaldin I., Lashkari A.H., Ghorbani A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *Proceedings of ICISSP*, 2018, pp. 108-116. DOI:10.5220/0006639801080116.
19. Canadian Institute for Cybersecurity, University of New Brunswick. CSE-CIC-IDS2018 Dataset. Electronic resource. Available at: `https://www.unb.ca/cic/datasets/ids-2018.html`
20. CERT Division, Software Engineering Institute, Carnegie Mellon University. Insider Threat Test Dataset (CERT), version 6.2. Electronic resource. Available at: `https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099`

Abdul Qayyum

Graduate student, ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 197101, Russia, St. Petersburg, 49 Kronverksky Ave., building A), e-mail: `qayyum068@gmail.com`. ORCID: 0009-0002-6226-0054.

Hamid Idris Mussa

M. Sc., ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 197101, Russia, St. Petersburg, 49 Kronverksky Ave., building A), e-mail: `haidturkey@outlook.com`.

Khalil Ibrahim

Graduate student, ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 197101, Russia, St. Petersburg, 49 Kronverksky Ave., building A), e-mail: khalil.ibrahim129@yahoo.com. ORCID: 0009-0000-4133-2923.

Sergey Valentinovich Bezzateev

D. Sc., Associate Professor, ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics, 197101, Russia, St. Petersburg, 49 Kronverksky Ave., building A); Saint-Petersburg State University of Aerospace Instrumentation e-mail: bsv@guap.ru. ORCID: 0000-0002-0924-6221.

Приватно-сохраняющее обнаружение вторжений в корпоративных сетях на основе федеративного обучения и дифференциально-приватного глубокого обучения

Абдул Кайюм[1], Хамид Идрис Мусса[1], Халил Ибрахим[1], С. В. Беззатеев[2]

[1]Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

[2]Санкт-Петербургский государственный университет аэрокосмического приборостроения

Аннотация: Централизованное обнаружение вторжений в распределённых корпоративных инфраструктурах (филиальные сети, удалённые офисы и промышленный/корпоративный IoT) создаёт ограничения по приватности и соблюдению требований (комплаенсу), поскольку требует централизации конфиденциальной телеметрии. В работе исследуется приватно-сохраняющий дизайн IDS на основе федеративного обучения (FL) с дифференциально-приватным локальным обучением (DP-SGD). Клиенты обучаются локально на признаках, извлечённых из сетевых потоков, и структурированной событийной телеметрии, а координатору для агрегации передаются только обновления модели. Мы оцениваем бюджет приватности (ε, δ) с использованием учёта приватности на основе RDP и качество обнаружения по метрикам Accuracy и Macro-F1. Эксперименты на CICIDS 2018 и наборе данных CERT Insider Threat v6.2 демонстрируют ожидаемый компромисс «приватность–полезность»: обучение с DP снижает полезность по сравнению с не-приватным централизованным обучением, а FL с DP обычно приводит к дополнительному снижению при гетерогенных (non-IID) разбиениях клиентов, оставаясь практичным при явных бюджетах приватности (основное сравнение: $\varepsilon \approx 1.8\text{-}2.0$). Также приводятся эксплуатационные показатели, релевантные для SOC: время обучения, пиковое потребление памяти, задержка вывода (inference) и размер модели, а также сравнение с базовыми моделями Random Forest и XGBoost.

Работа выполнена в рамках государственного задания (проект FSER-2025-0003).

Ключевые слова: обнаружение вторжений, федеративное обучение, дифференциальная приватность, DP-SGD, безопасная агрегация, инсайдерские угрозы, информационная безопасность, корпоративные сети, центр мониторинга и реагирования на инциденты (SOC).

Для цитирования: Кайюм Абдул, Мусса Хамид Идрис, Ибрахим Халил, Беззатеев С. В. Приватно-сохраняющее обнаружение вторжений в корпоративных сетях на основе федеративного обучения и дифференциально-приватного глубокого обучения // Вестник СибГУТИ. 2026. Т. 20, № 2. С. 3–15. https://doi.org/10.55648/1998-6920-2026-20-2-3-15.

Статья поступила в редакцию 27.02.2026;
переработанный вариант – 16.04.2026;
принята к публикации 16.04.2026.

Кайюм Абдул
аспирант, Университет ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49, лит. А), e-mail: qayyum068@gmail.com. ORCID: 0009-0002-6226-0054.

Мусса Хамид Идрис
магистр, университет ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49, лит. А), e-mail: haidturkey@outlook.com.

Ибрахим Халил
аспирант, университет ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49, лит. А), e-mail: khalil.ibrahim129@yahoo.com. ORCID: 0009-0000-4133-2923.

Беззатеев Сергей Валентинович
д-р техн. наук, доцент, доцент, Университет ИТМО (Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49, лит. А), e-mail: bsv@guap.ru. ORCID: 0000-0002-0924-6221.